Sworn Declaration to be presented in Patent Prosecution Proceedings

I, the undersigned, Martin Lehnert, having been informed that a false sworn declaration is subject to prosecution as a criminal offence under § 156 of the German Penal Code, hereby make the following sworn deposition which I affirm to be true and accurate to the best of my knowledge.

I refer to numbered point 5 of the Examiner's letter dated 3rd November 2003 on my US application 09/729,898. The mentioned point is valid for applications, which demand exact handling of numbers, for example for financial transactions. In this case, the bit-width is normally high, the timing requirements are moderate, the value of a digit well-defined, and manipulation of a number is not an option.

There are, however, different applications, where the input data yield a limited resolution and thus include tolerances, the used bit-width is limited and the time constraints for signal processing are significant due to real-time application. An example for this very widespread type of applications is the analogue to digital (A/D) conversion of input signals and use of the resulting digital numbers for calculations in a subsequent Microprocessor (MP) or Digital-Signalling-Processor (DSP). Here my method can yield significant advantages for operation and the result of calculation does either not differ from the result of state-of-the-art number extension or the difference does not impact the operation of the system in a negative way or the difference improves the overall system performance, verified by simulations.

For an example let's assume the mentioned example, an A/D-converter delivering a 4-digit word in 2's-complement notation (range: -8,...,7) connected to a 16-bit DSP using 16 digits for further processing. The A/D-converter has a limited resolution, which is in best-case ½ digit. Thus if the A/D-converter delivers the decimal value 2, binary (0010), in fact the input value can be any value between 1.5 and 2.5. Thus extending the value (0010) by 0, as in state-of-the-art to 0010 000000000000 is as correct or wrong as extending it to 0010000000000001, because the additional digits are located inside the tolerance range of the A/D-converter and thus unknown. The input values of A/D conversion are preferably assigned to the most-significant-bits (MSB) of a word, because otherwise precision of the input data is lost, or additional hardware maybe required to extend the sign bit towards the most significant digits. In many cases a multiplication (MPY) of the input data occurs. This may be necessary to scale the input data to an appropriate range, calculate the input energy by MPY of the sample by itself or due to a variety of other reasons. In the following description the 2-s complement notation is used, which is the most common format in digital signal processing. The result of the MPY of the input data (0010) by (0001000000000000) and consecutive handling in the DSP is shown.

1	Input from A/D-conversion:	(0010)
2	Input to DSP (with my method):	(0010000000000001)
3	Multiply by:	(0001000000000000)
4	Result of multiplication:	(00001000000000000001000000000000)
5	Extract MSB 16 digits for further operation:	(0000100000000000)

After the multiplication, the result must be limited to 16-bit to avoid that the bit-width doubles after each multiplication. One method is to extract the 16 most significant bits and the 1 bit, added according to my method in the least significant bit here, disappears in the result word and the value is the correct, expected one. In the example the input word (0010) is connected to the most significant bits, which is the common method to gain the maximum achievable precision for the multiplication.

It can be shown that for appropriate choice of the multiplication factor and position of added "1" digit, the correct result, expected due to mathematics, is maintained. Thus my method does not deliver incorrect results and is therefore applicable.

If the location of the additionally inserted "1" would be chosen differently from the least significant bit, such that the result of multiplication is impacted by this "1", this may be tolerable or even desirable.

There are cases that during further processing the affected part of the 16-bit-word will be cut off. There are also applications, where a different result due to use of my method is intended, for example to improve the statistical properties of the input signal (with virtually no additional effort). In this case simulation of the complete operation chain (A/D conversion → scaling in DSP → post processing) can verify improved overall operation, when applying my method.

However, one obvious, significant advantage of my method is that it solves a problem of signal processing resulting from the used number format. In the uttermost cases of digital signal processing the 2's complement is used to represent digital numbers. The 2's complement is by definition non symmetrical, which means that the range of numbers >0 is by one smaller than the range of numbers <0. Using the example above, a 4-digit binary number in 2's complement can represent the numbers decimal -8 (binary 1000) up to decimal +7 (binary 0111), which are 7 numbers >0, but 8 numbers <0. The maximum negative number (in the example –8) has no counterpart in the positive domain. This causes the effect that if the most negative number is multiplied by the most negative number ((-8)*(-8)) the result exceeds the available 2's complement number range. If no further actions are taken the result cannot be represented in the given range, which may lead to the incorrect output of hardware: (-8)*(-8)=(-64)??? There are several ways to handle this issue, but these normally yield disadvantages, like additional computational complexity, which is a problem in real-time applications. My method presents an effective way to avoid these disadvantages by adding a small increment to the input number, thus avoiding the occurrence of the maximum negative number, with effectively no additional hardware effort. My method also yields advantages and applications in other scenarios, as given above and in the description of my patent application.

As shown above, my method is applicable and yields advantages for operation in the application described in the given example and also for other applications.

I hereby declare and affirm that the foregoing statement is true and accurate to the best of my knowledge.

Nuremberg, 17 December 2003

Martin Lehnert